Produktentwicklung 2

TA.BA\_PREN2.FS14

**Spezifikation: BluetoothController-Klasse**

Version 1.0.0 (FS 2015) / 15. April 2015

Simon Neidhart

## Inhalt

[1 Steckbrief 1](#_Toc5199)

[2 Operationen und Datenstukturen 2](#_Toc5200)

[3 Einsatz, Abläufe, Voraussetzungen und Zusicherungen 2](#_Toc5201)

[4 Aufbau und Konfiguration 2](#_Toc5202)

[5 Fehlerbehandlung 2](#_Toc5203)

[6 Qualitätsmerkmale 2](#_Toc5204)

[7 Entwurfsentscheidungen 3](#_Toc5205)

[8 Beispielverwendung 3](#_Toc5206)

[9 Änderungsmanagement 3](#_Toc5207)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Version* | *Datum* | *Autor* | *Bemerkung* | *Status* |
| 1.0.0 | 26.03.2015 | SN | Initiale Fassung | freigegeben |
| 1.0.1 | 15.05.2015 | SN | Initiale Fassung | freigegeben |

# Steckbrief

|  |  |
| --- | --- |
| *Name* | **BluetoothController** |
| *Beschreibung* | Definiert eine Klasse, um Nachrichten an ein ausgewähltes BluetoothGerät zu übermitteln und stellt die Möglichkeit zur Verfügung, auf demselben seriellen Port Nachrichten zu empfangen. |
| *Typ* | Java Class |

Tabelle 1: Steckbrief der Klasse BluetoothController

# Operationen und Datenstukturen

Die Schnittstelle stellt folgende Methoden und Datentypen zur Verfügung:



Abbildung 1: UML Klassendiagramm der Klasse BluetoothController

Details zu den Methoden und den verwendeten Datentypen sind in der JavaDoc festgehalten.

# Einsatz, Abläufe, Voraussetzungen und Zusicherungen

* Bevor Daten über die Schnittstelle ausgetauscht werden können, muss mittels initConnections(String serialPortName) ein gültiger serieller Port definiert werden.
* Der konfigurierte Port gilt für alle darauf folgenden command-Aufrufe.
* Ein Wechsel des seriellen Ports, d.h. Umkonfigurierung mit initConnections(String serialPortName), ist nun jederzeit möglich.
* Eine Klasse, die das Interface BluetoothReceiverListener implementiert, kann als Observer für empfangene Nachrichten des BluetoothReceivers fungieren.

# Aufbau und Konfiguration

Keine zusätzlichen Informationen.

# Fehlerbehandlung

Die Fehlerbehandlung wird über unchecked Exceptions realisiert. Details siehe JavaDoc.

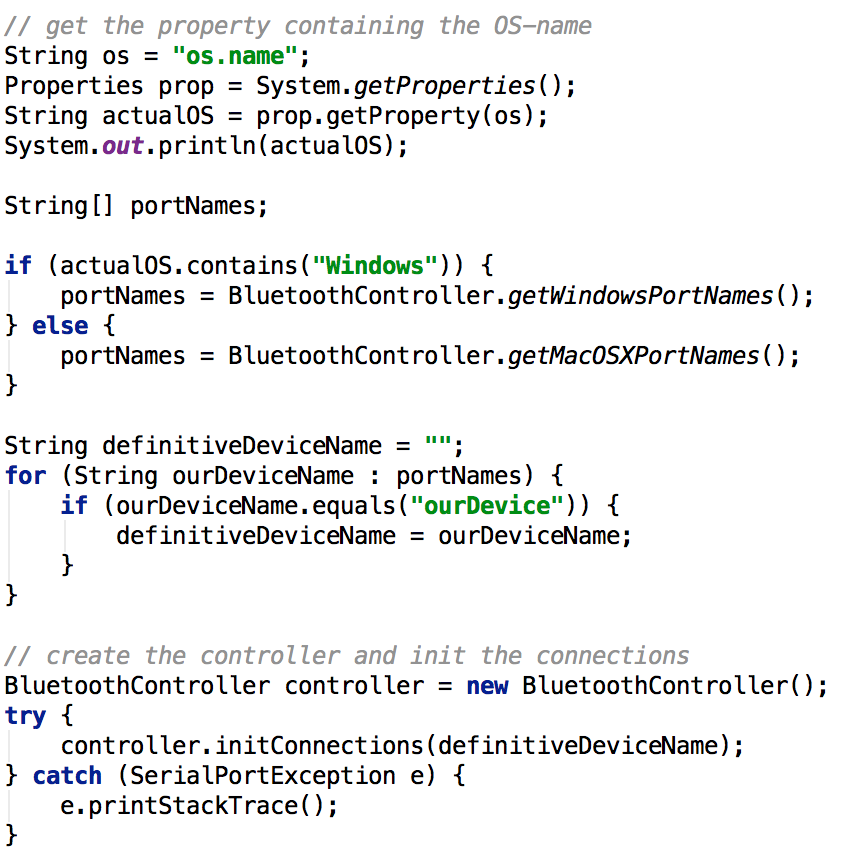
# Qualitätsmerkmale

# Entwurfsentscheidungen

* .
* .
* .

# Beispielverwendung

Der folgende Codeausschnitt zeigt die Verwendung der Schnittstelle anhand einer beispielhaften Implementation PersistedStringCsv:



# Änderungsmanagement

Änderungswünsche an diese Schnittstellendefinition oder an deren Implementation können mit entsprechender Begründung per Mail an das Dozierendenteam eingereicht werden. Von diesem wird dann entschieden (ggf. im Plenum) ob die Änderungswünsche erfüllt werden.